Building a consumer anonymity scalable payment protocol for the Internet purchases

H. Wang J. Cao
Department of Maths & Computing
University of Southern Queensland
Toowoomba QLD 4350 Australia
(wang, cao)@usq.edu.au
Yahiko Kambayashi
Department of Social Informatics
Kyoto University
Kyoto, Japan
yahiko@db.soc.i.kyoto-u.ac.jp

Abstract

This paper proposes a secure, anonymity scalable and practical payment protocol for the Internet purchases. It uses electronic cash for payment transactions. In this new protocol, users can improve the anonymity for themselves if they worry about disclosure of their identities. We delegate an agent to provide the higher anonymous certificate and improve the security. The agent will certify re-encrypted data after verifying the validity of the content from users, but without any private information of the users. With this new method, each user can get the required anonymity level, depending on the available time, computation and the cost. Furthermore, the new payment protocol can prevent from eavesdropping, tampering and impersonation effectively.

We also analyse how to prevent a user from spending a coin more than once and discuss how to use the proposed protocol for the Internet purchases. The new method provides an efficient and untraceable electronic cash scheme. It is promising for electronic trades through the Internet.

Keywords: Electronic-cash, Anonymity, Traceability, Hash function.

1 Introduction

Recent advances in the Internet and WWW have enabled rapid development in E-Commerce. More and more business begin to develop or adopt E-commerce systems to support their selling activities. While this brings convenience for both consumers and vendors many users have concerns about the security and their private information when purchasing over the Internet. Especially, with the electronic payment or E-case payment. Users often prefer to have some degrees of anonymity when doing shopping over the Internet. There are a number of proposals for electronic cash systems. All of them lack of flexibility in anonymity. David Chaum [5] first proposed an on-line payment system that will guarantee receiving valid coins. This system provides anonymity against a collaboration

of shops and the bank. Another on-line CyberCoin (http://www.cybercash.com) approach allows clients making payments by signing fund transfer requests to merchants. The merchants submit the signed requests to the bank for authorization of the payments. The protocol is not fully anonymous since it allows the issuing bank to track every purchase. Furthermore, the scalability of the CyberCoin protocol is questionable since it relies on the availability of a single on-line bank. NetBill [8] extends the above payment mechanism by supporting goods atomicity and

certified delivery. The drawbacks of its protocol are the addition of extra messages, and the significant increase in the amount of encryption used. However, DigiCash [7] uses blind signatures to provide a fully anonymous coin-based payment system. This system has the disadvantages of centralized management of issuing and checking double spending of coins.

The most sophisticated protocol is the SET protocol [12], which was designed to facilitate credit card transactions over the Internet. SET security comes at a considerable computation and communication cost. SET, unlike other simpler on-line protocols, does not offer full anonymity, non-repudiation or certified delivery.

The above systems are on-line payment systems. They need sophisticated cryptographic functions for each coin, and require additional computational resources for the bank to validate the purchases. Forcing the bank to be on-line at payment is a very strict requirement. On-line payment systems secure the merchant and the bank against customer fraud, since every payment needs to be approved by the customer's bank. The primary disadvantage of on-line authorization is the cost associated with per transaction, imposed by requirements for a highly reliable and efficient clearing system at the customer's bank.

In an off-line protocol, the merchant verifies the payment using cryptographic techniques, and commits the payment to the payment authority later in an off-line batch process. Off-line payment systems were designed to lower the cost of transactions due to the delaying to verify the batch processes. Off-line payment systems, however, suffer from the potential of double spending, whereby the electronic currency might be duplicated and spent repeatedly.

The first off-line anonymous electronic cash was introduced by Chaum, Fiat and Naor [7]. The security of their scheme relied on some restrict assumptions. There is also no formal proof was attempted. Although hardly practical, their system demonstrated how off-line e-cash can be constructed and laid the foundation for more secure and efficient schemes.

1995, Chan, Frankel and Tsiounis [4] presented a provable secure off-line e-cash scheme that relied only on the security of RSA [16]. This scheme extended the work of Franklin and Yung [11] who aimed to achieve provable security without the use of general computation protocols. The anonymity of users is based on the security of RSA and cannot be changed after the system established. NetCents [15] proposed a lightweight, flexible and secure protocol for micropayments electronic commerce over the Internet. This protocol is designed only to support purchases ranging in value from a fraction of a penny and up. In 2000, David Pointcheval [14] presented a payment scheme, in which the user's identity can be found any time by the certification authority. So the privacy of a user cannot be protected.

As mentioned above, the on-line e-cash payments need much more computing resources. The most proposed off-line schemes are only for micropayments. They just rely on the heuristic proofs of security and therefore do not formally prevent fraud and counterfeit money. Furthermore, most on-line and off-line payment schemes do not provide efficient anonymity for users. Hence, a new payment scheme for the purchases over the Internet with both untraceable and flexible anonymity

will be very useful and very important.

In this paper, we first analyse e-payment models, then propose a new off-line electronic cash scheme, in which the anonymity of users is scalable and that can be done by user themself. Users can get the required anonymity without showing their identities to any other third part. This is a true anonymous for the legal users and it can trace users' identities for double spending.

The paper is organized as follows. In the following section, some basic definitions and the simple examples are reviewed. The payment model and the anonymity provider agent are described in section 3. The design of a new off-line electronic cash scheme and its complexity are detailed in section 4 and the security analysis of our scheme is given in section 5. How to use the new e-cash for the Internet purchases is given in section 6. Conclusions are included in section 7.

2 Some Basic Definitions

2.1 Hash functions

$h(x)$ is a hash function. For a given value x it is computationally hard to find a $y \neq x$ such that $h(x) = h(y)$, i.e., collisions are hard to find.

Hash function is a major building block for several cryptographic protocols, including pseudo-random generators [1, 2], digital signatures [3], and message authentication.

2.2 DLA and ElGamal encryption system

Discrete Logarithm Assumption (DLA) is an assumption that the discrete logarithm problem is believed to be difficult and also to be the hard direction of a one-way function.

The discrete logarithm problem is as follows: given an element g in a group G of order t, and another element y of G, find a x, where $0 < x < t - 1$, such that y is the result of composing g with itself x times. In some groups there exist elements that can generate all the elements of G by exponentiation (i.e., applying the group operation repeatedly) with all the integers from 0 to $t - 1$. When this occurs, the element is called a generator and the group is called cyclic. Rivest [17] has analyzed the expected time to solve the discrete logarithm problem both in terms of computing power and cost.

For this reason, it has been used for the basis of several public-key cryptosystems, including the famous ElGamal encryption system. ElGamal encryption system [9] is a public key encryption scheme which meets the semantic security. Let us briefly recall it.

step 1. The system needs a group $\Im$ of order q, and a generator g. The secret key is an element $X \in Z_q$ and the public key is $Y = g^X$. step 2. For any message $m \in \Im, c = \varepsilon(Y, m; r) = (g^r, Y^r m)$, for a random $r \in Z_q - \{0\}$. step 3. For any ciphertext $c = (a, b), m = D(X, c) = b/a^X$.

ElGamal encryption scheme

2.3 Undeniable signature scheme and Schnorr signature scheme

Undeniable signature scheme, devised by Chaum and van Antwerpen [6], is non-self authenticating signature schemes, where signatures can only be verified with the signer's consent. However, if a signature is only verifiable with the aid of a signer, a dishonest signer may refuse to authenticate a genuine document. Undeniable signatures solve this problem by adding a new component called the disavowal protocol in addition to the normal components of signature and verification.

An undeniable proof scheme consists of the following algorithms:

1. The key generation algorithm K which outputs random pairs of secret and public keys (sk, pk).

2. The proof algorithm $P(sk, m)$ which, on input a fact m, returns an "undeniable signature" s on m.

However this proof "s" does not convince anybody by itself. To get convinced of the validity of the pair (m, s), relatively to the public key pk, one has to interact with the owner of the secret key sk.

3. The confirmation process confirmation (sk, pk, m, s) which is an interactive protocol between the signer and the verifier,where the prover(the signer) tries to convince the validity of the pair (m, s).

4. The disavowal process disavowal (sk, pk, m, s) which is an interactive protocol between the signer and the verifier,where the prover (the singer) tries to convince that the pair (m, s) is not valid (i.e. has not been produced by him).

Schnorr proposed a undeniable signature scheme in 1991 [18]. We simply recall it.

The system needs primes p and q such that $q\|p-1$, $g \in Z_p$ with order q, i.e. $g^q = 1(modp), g \neq 1$. A user generates by himself a private key s which is a random number in Z_q. The corresponding public key v is the number $v = g^{-s}(modp)$.
To sign message m with the private key s perform the following steps: 1. Compute $x = g^r(modp)$, where $r \in Z_q$ is a random number. 2. Compute $e = h(x, m)$, where h is a hash function. 3. Compute $y = r + se(modq)$ and output the signature (e, y).
To verify the signature e, y for message m with public key v compute $\overline{x} = g^y v^e(modp)$ and check that $e = h(\overline{x}, m)$.

Schnorr signature scheme

3 Basic model and new payment model

We will discuss the basic payment model and our new payment model in this section.

3.1 Basic payment model

Electronic cash has sparked wide interest among cryptographers ([10, 21, 17, 22, 13], etc.). In its simplest form, an e-cash system consists of three parts (a bank B, a user U and a shop S) and three

main procedures as shown in Figure 1 (withdrawal, payment and deposit). In a coin's life-cycle, the user U first performs an account establishment protocol to open an account with the bank B.

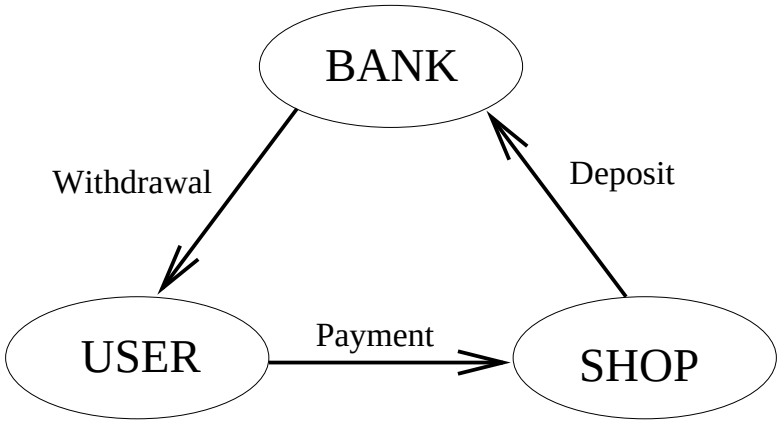

Figure 1: Basic electronic cash system

The users and the shops maintain an account with the bank, while

1. U withdraws electronic coins from his account, by performing a withdrawal protocol with the bank B over an authenticated channel.
2. U spends a coin by participating in a payment protocol with a shop S over an anonymous channel, and
3. S performs a deposit protocol with the bank B, to deposit the user's coin into his account.

The system is *off-line* if the shop S does not communicate with the bank B during payment. It is *untraceable* if there is no p.p.t. TM (probabilistic polynomial-time Turing Machine) that can identify a coin's origin even if it accesses to all withdrawal, payment and deposit transactions. It is *anonymous* if the bank B, in collaboration with the shop S, cannot trace the coin to the user. However, in the absence of tamper-proof hardware, electronic coins can be copied and spent multiple times by the user U. This has been traditionally referred to as double-spending. In on-line e-cash, double-spending is prevented by having the bank check if the coin has been deposited before. In off-line e-cash, however, this solution is not possible; instead, as proposed by Chaum, Fiat and Naor [7], the system guarantees that if a coin is double-spent the user's identity is revealed with overwhelming probability.

There are also three additional processes such as the bank setup, the shop setup, and the user setup (account opening). They describe the system initialization, namely creation and posting of public keys and opening of bank accounts. Although they are certainly parts of a complete system, these are often omitted as their functionalities can be easily inferred from the description of the three main procedures. For clarity we will only describe the bank setup and the user setup (because the shop setup is as similar as the user setup) for our new scheme in the next section.

Besides the basic participants, a third part named Anonymity Provider (AP) agent will be involved in our scheme. AP agent will help the user to get required anonymity and it will not be involved in a purchase process. The new model can be shown in Figure 2. The AP agent gives a certificate to the user when s/he needs a higher level anonymity.

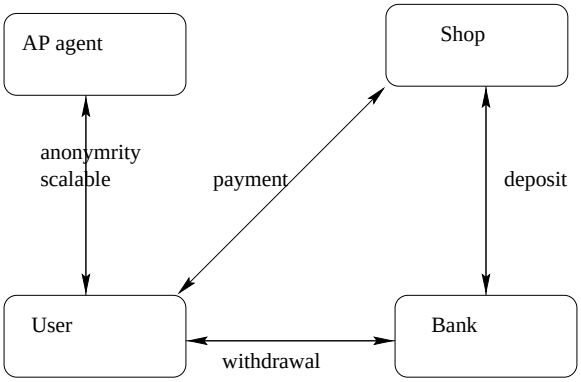

Figure 2: New electronic cash model

3.2 Anonymity Provider Agent

Here we explain what the AP agent will do.

We assume that a user owns a valid coin $c = \Im(pk_B, pk_u, y)$ with its certificate $Cert_c$, which guarantees correct withdrawal from the bank. Whether a coin is valid or not is depending on its certificate. Therefore the bank can do revocation if found a user who spends a coin twice. After the following processes with the AP agent, the user owns a new valid coin, $c' = \Im(pk_B, pk_u, y + t)$ with its certificate $Cert_{c'}$.

1. The user re-encrypts the coin c into $c' = \Im(pk_B, pk_u, y + t)$.
2. The user provides an undeniable signature S, using c as a public key associated with the secret key (sk_u, y), of the equivalence between c and c'. This equivalence is guaranteed by the existence of t.
3. The user confirms the validity of this signature S to the AP agent.
4. The AP agent certifies the new coin c' and sends $Cert_{c'}$ to the user.

Indeed, after steps 2 and 3, the AP is convinced of that the conversion has been performed by the owner of the coin c; c' is equivalent to c. The owner of c will not be able deny later S (the relation between c and c').

3.3 Proof of ownership of a coin

Let us assume that Y is the public key of the bank, and $I = g^{x_u}$ the identity of a user. $H(x, y)$ is a hash function. A coin is an encryption of I: $c = (a = g^r, b = Y^r I^s)$ which is afterwards certified by the bank, where r, s are random numbers. With the certificate of the bank, one knows that the encryption is valid. Therefore, in order to prove his ownership, the user has just to convince of his knowledge of (x_u, r, s) such that $b = Y^r I^s$. This can be expressed in Figure 3.

1. The prover chooses random $k \in Z_q$ and computes $t = Y^k g^{s+x_u se-x_u e}$.
2. He then computes $u = k - re(modq)$ and $v = s - x_u e(modq)$, where $e = H(m, t)$ and m is a message.
3. The signature finally consists of the triple (e, u, v).
4. In order to verify it, one has just to compute $t' = Y^u g^v b^e$ and check whether $e = H(m, t')$ or not.

Figure 3. Proof of validity of a coin $c = Y^r I^s$

Then, a scrambled coin is simply got by multiplying both parts of the old one by respective bases, g and Y, put at a same random exponent ρ :

$$c' = (a' = g^\rho a, b' = Y^\rho b) = (g^{r+\rho}, Y^{r+\rho} I^s).$$

Then, if the owner of the old coin has certified the message $m' = h^\rho$, equivalence of both coins can be proven with the proof of equivalence of three discrete logarithms:

$$log_h m' = log_g(a'/a) = log_Y(b'/b).$$

4 Anonymity self-scalable payment scheme

In this section, we propose an anonymity self-scalable payment scheme. The new payment scheme has two main features, one is that a user can get higher level anonymity by itself, another one is that the identity of a user can not be traced except the user spends the same coin twice.

Our scheme includes two basic processes in system initialization (bank setup and user setup) and three main protocols: a new withdrawal protocol with which U withdraws electronic coins from B while his account is debited; a new payment protocol with which U pays the coin to S; and a new deposit protocol with which S deposits the coin to B and has his account credited. If a user wants to get higher level anonymity after s/he got a coin from the bank (withdrawal), s/he can contact to the AP agent .

4.1 System Initialization

The bank setup and the user setup based on Discrete Logarithm Assumption are described as follows, and the details of the shop setup are omitted (because the shop setup is similar to the user setup).

Bank's setup: (performed once by B)
Primes p and q are chosen such that $|p-1| = \delta + k$ for a specified constant δ, and $p = \gamma q + 1$, for a specified small integer γ. Then a unique subgroup G_q of prime order q of the multiplicative group Z_p and generator g of G_q are defined. Secret key $x_B \in_R Z_q$ for a denomination is created, where $a \in_R A$ means that the element a is selected randomly from the set A with uniform distribution. Hash function H from a family of collision intractable (or, ideally, according to [10], correlation-free one way) hash function is also defined. B publishes p, q, g, H and its public keys $Y = g^{x_B} (\mathrm{mod}\ p)$.

The secret key x_B is safety under the DLA. The Hash function will be used in payment transactions.

User's setup (account opening): (performed for each user U)
The bank B associates the user U with $I = g^{x_u} (\mathrm{mod}\ p)$ where $x_u \in G_q$ is the secret key of the user and is generated by U.

In system initialization, the communication complexity is $O(1)$ for the user only sends its account I of length l bits to the bank, and the computation complexity is $O(1)$. It requires only two exponentiations g^{x_B} and g^{x_u}.

After the user's account and the shop's account opening, we can describe the new payment scheme.

4.2 New off-line payment scheme

We now describe the new anonymity scalable electronic cash scheme which includes withdrawal, payment and deposit.

Withdrawal: As usual, an anonymous coin is a certified message, which embeds the public key of a user. In our scheme, the message is an encryption of this user's public key, using the public key Y of the bank.

Instead of using intricate zero-knowledge proofs to convince the bank of the validity of the encryption, the user shows some information to the bank including a signature. So the bank certifies the encryption with full confidence.

The user $I = g^{x_u}$ constructs a coin $c = (a = g^r, b = Y^r I^s)$ using the public key Y of the bank, where s is a secret key of the coin, which is kept by the user and r is a random number in Z_q. He also signs c together with the date, using his private key x_u and a Schnorr signature. He sends both to the bank together with r, I. Then the bank can check the correct encryption. With the signature of the coin and the date, only the legitimate user could have done it. After having modified the user's account, the bank sends back a certificate $Cert_c$. The user just has to remember $(r, s, Cert_c)$.

Anonymity scalability: The user can use the coin now without higher anonymity since the bank can easily trace any transaction performed through the coin. This is because some information of the user such as $I, Cert_c$ have been known by the bank. To solve this problem, an AP agent is established to help the user making the coin anonymous: the user can derive a new encryption of his identity in an indistinguishable way. However, the user will need a new certificate for a new issued ciphertext. The AP agent can provide this new certificate. Before certifying, the user requires both the previous coin $(c, Cert_c)$ and the proof of equivalence between the two ciphertexts. Details are described as below.

The user contacts to the AP agent if s/he needs to get higher level anonymity. The user chooses a random ρ and re-encrypts the coin:

$$c' = (a' = g^\rho a, b' = Y^\rho b).$$

1. The user generates a signature $S = (e, u, v) = \Im((r, x), m)$ on $m = h^\rho$ using the secret key (r, x) associated with the public one $b = Y^r I^s$ as shown on Figure 3. Because of S, the user will not be able to deny his knowledge of ρ later. Furthermore, nobody can impersonate the user at this step, since the discrete logarithm x_u of I is required to produce a valid signature. So there is no existential forgery.

2. The user also provides a designated -verifier proof of equality of discrete logarithms

$$log_h m = log_g(a'/a) = log_Y(b'/b).$$

3. The user finally sends c, c', S, m to the AP agent.

4. The AP agent checks the certificate $Cert_c$ on c, the validity of the signature S on the message m using the public key b. He then certifies c' and sends back the certificate $Cert_{c'}$ to the user.

After these processes the user gets a new certified coin $c' = (a' = g^{\rho}a, b' = Y^{\rho}b, Cert_{c'})$ which is now strongly anonymous from the point of view of the bank. The AP agent has to keep (c, c', m, S) to be able to prove the link between c and c', with the help of the user. Users need to send I to the AP agent.

In withdrawal process, the communication complexity is $O(1)$ since the user sends c, I and a signature to the bank and the bank returns $Cert_c$ to the user, seven exponentiations are required in withdrawal and anonymity providing process.

Follow the process, the AP agent can also give many smaller new coins for an old one since the amount of new one can be embedded in the certificate $Cert_{c'}$.

Payment: (performed between the user and the shop over an anonymous channel)
When a user possesses a coin, he can simply spend it at shops: he proves his knowledge of the secret key (x_u, s) associated with the coin c or c'. This proof is a signature $S = (e, u, v)$ of the purchase, date, etc with the secret key x, s associated to the coin to the receiver (which is later forwarded to the bank).

In payment transactions, the communication complexity is $O(1)$ for the user sending c and a signature $S = (e, u, v)$ to the shop. There are five exponentiations for the signature.

Deposit:(The receiver deposits a coin to a bank)
Since the system is off-line, the shop will send the payment transcript to the bank B later. The transcript consists of the coin c or c' (if the user applied high level anonymity), the signature and the date/time of the transaction. The bank will verify the correctness of payment and credit coin into shop's account.

In deposit, the communication complexity is $O(1)$ because the shop sends user's response c, and signature $S = (e, u, v)$ to the bank. The computation complexity is $O(1)$, since it only verifies whether c or c' was used before or not.

Untracebility: The receiver (shop) deposits the coin into its bank's account with a transcript of the payment. If the user uses the same coin c twice, then the user will be traced: two different receivers will send the same coin c to the bank. The bank can easily search its records to ensure that c has not been used before. If the user uses c twice, then the bank has two different signatures. Thus, the bank can isolate the user and trace the payment to the user's account I.

In our new scheme, the communication complexity is $O(1)$ and fourteen exponentiations are required. So it is quite efficiency.

5 Security Analysis

An off-line E-cash scheme is secure [11] if the following requirements are satisfied:

1. *Unreusable*: If any user uses the same coin twice, the identity of the user's can be computed.

2. *Unexpandable*: With n withdrawal processes, no p.p.t. (Probabilistic polynomial time) Turing Machine can compute $(n+1)$th distinct and valid coin.

3. *Unforgeable*: With any numbers of the customer's withdrawal, payment and deposit, no p.p.t. Turing Machine can compute a single valid coin.

4. *Untraceable*: With any numbers of the customer's valid withdrawal, payment and deposit protocols, no p.p.t. Turing Machine can compute a legal user's identity.

The security in our e-cash scheme is based on the hardness of Discrete Logarithms [23] and hash functions that preserves the above four requirements.

Unreusable: When a user spending a coin, he hands over the coin together with a signature $S = (e, u, v)$ to a shop. If the user uses a coin twice, then we have two signatures $S_1 = (e_1, u_1, v_1)$ and $S_2 = (e_2, u_2, v_2)$, where

$$u_1 = k_1 - re_1(modq),\ v_1 = s - xe_1(modq).$$

$$u_2 = k_2 - re_2(modq),\ v_2 = s - xe_2(modq).$$

Then $(v_2 - v_1)/e_1 - e_2 = x$, this is the secret key of the user I. So, a coin in the new scheme cannot be reused.

Untraceable: When a user constructs a coin, s/he uses the secret key x and s, both of them are not shown to any other parts in the purchase process. So no one can trace the user and the coin.

Unforgeable: As already seen, the secret key x of a user is never revealed, but only used in some signatures. Any user is therefore protected against any impersonation, even from a collusion of the bank, the AP agent, and the shop. Only the user can construct a valid coin since there is a undeniable signature embedded in the coin. To prevent the bank frame the user as a multiple spender in the scheme, we use digital signature I^s for s is known only by the user. The user is protected against frame-up only computationally, not unconditionally.

Unexpandable: For a legal user and a valid coin, the secret key x and the random number s never shown others in anytime. Further more, as usually, the random number s will be changed for different coins. With n withdrawal proceedings, the random number s will be changed n times. Then, no one can compute $(n+1)th$ distinct and valid coin even see n withdrawal proceedings.

6 Implementation of the Internet purchases

In this section, we analyse two different purchase procedures. We will show how to use the new e-cash for the Internet purchases and we will see the efficiency of the payment protocol.

Purchase procedure 1

We assume the purchase procedure 1 is that: the user decides how much money should be paid to the shop and withdraws the money from the bank, then pays it to the shop.

1. *User to shop*: The user wants to buy some goods in a shop, s/he contacts to the shop for the price.

2. *User to bank*: The user gets the amount of money form the bank, the amount is embedded in the signature.

3. *Anonymity scalability*: If the user wants to get higher anonymity, s/he can ask AP agent to certify a new cash. S/he can use the new cash to the shop.

4. *User to shop*: The user proves to the shop that s/he is the owner of the money and pays it to the shop. Then the shop sends the goods to the user.

5. *Shop to bank*: The shop deposits the e-cash to the bank, the bank checks the validation and no double spending of the coin. The bank increases the money into the shop's account.

Purchase procedure 2

We assume the purchase procedure 2 is that: the user does not have to ask the bank sending money since the user has enough e-cash in her/his "wallet". All s/he needs to do is that s/he should get some smaller e-cash to pay the shop. This means the user can keep e-cash itself.

There are also 4 steps in the purchase procedure 2, they are: *user to shop*; *user to AP agent*; *user to shop* again and *shop to bank*. Only *user to AP agent* is different from the procedure 1 and another three steps are the same. We just describe *user to AP agent*.

User to AP agent: The user prepares the amount of money to pay the shop from his/her wallet. S/he can ask AP agent to make some smaller coins. In the same time the user might get higher anonymity. After checking the old money, the AP agent creates some new e-cash which the total is equivalent to the old e-cash. One of the coins is paying the shop.

We have already seen that the user can keep money in its wallet or get money from the bank. And in both purchase procedures 1 and 2, most computations are done by the users. It is very convenient for the Internet purchases.

7 Conclusions

In this paper, a new electronic cash scheme is designed to provide different degrees of anonymity for users. Users could decide the anonymity levels. They can have low anonymity if users want to spend coins directly after withdrawing coins from the bank. Users can get higher level anonymity through the AP agent without showing their private information. Users are more secure from the bank's point of view because the new certificate of a coin comes from the AP agent which does not involve in a payment process. It does not need a trusted part to manage users' identities. We have shown how to derive an efficient and untraceable cash scheme based on the variants of coins in the new model. It is an off-line scheme with low communication and computation. With this scalable anonymity, the new payment protocol can prevent from eavesdropping, tampering and impersonation effectively. Finally, we have discussed how to use the new e-cash over the Internet.

References

[1] Bellare M., Goldreich O., and Krawczyk H. Stateless evaluation of pseudorandom functions: Security beyond the birthday barrier. In *Advances in Cryptology - Crypto 99*, volume 1666 of *Lectures Notes in Computer Science*. Springer-Verlag, 1999.

[2] Boyko V., Peinado M., and Venkatesan R. Speeding up Discrete Log and Factoring Based Schemes via Precomputations. In *Advances in Cryptology - Eurocrypt'98*, volume 1807 of *Lectures Notes in Computer Science*. Springer-Verlag, 1998.

[3] Canetti R., Goldreich O., and Halevi S. The Random Oracle Methodology. In *Proceedings of the 30th ACM STOC '98*, pages 209–218. IEEE, 1998.

[4] Chan A., Frankel Y., and Tsiounis Y. An efficient off-line electronic cash scheme as secure as RSA. Research report nu-ccs-96-03, Northeastern University, Boston, Massachussets, 1995.

[5] Chaum D. Blind signature for untraceable payments. In *Advances in Cryptology - Crypto 82*, pages 199–203. Plenum Press N.Y., 1983.

[6] Chaum D. and Van antwerpen H. Undeniable signatures. In *Advances in Cryptology–Crypto89*, volume 435 of *Lectures Notes in Computer Science*, pages 212–216. Springer-Verlag, 1990.

[7] Chaum D., Fiat A., and Naor M. Untraceable electronic cash. In *Advances in Cryptology - Crypto 88*, volume 403 of *Lectures Notes in Computer Science*, pages 319–327. Springer-Verlag, 1990.

[8] Cox B., Tygar J.D., Sirbu M. Netbill security and transaction protocol. In *The first USENIX workshop on electronic commerce*, New York, 1995.

[9] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on information theory*, IT-31(4):469–472, 1985.

[10] Frankel Y., Yiannis T., and Yung M. Indirect Discourse Proofs: Achieving Fair Off-Line Electronic Cash. In *Advances in cryptology–Asiacrypt'96*, volume 1163 of *Lectures Notes in Computer Science*, pages 286–300. Springer-Verlag, 1996.

[11] Franklin M., Yung M. Secure and efficient off-line digital money. In *Proc. of the twentieth International Colloquium on Automata, Languages and Programming*, volume 700 of *Lectures Notes in Computer Science*, pages 265–276. Springer-Verlag, 1993.

[12] MastercardVisa, editor. *SET 1.0 - Secure Electronic Transaction Specification*. http://www.mastercard.com/set.html, 1997.

[13] Okamoto T. An efficient divisible electronic cash scheme. In *Advances in cryptology–Crypto'95*, volume 963 of *Lectures Notes in Computer Science*, pages 438–451. Springer-Verlag, 1995.

[14] Pointcheval D. Self-scrambling anonymizers. In *Proceedings of Financial Cryptography*, Anguilla, British West Indies, 2000. Springer-Verlag.

[15] Poutanen T., Hinton H. and Stumm M. Netcents: A lightweight protocol for secure micropayments. In *The 3rd USENIX workshop on electronic commerce*, Boston, Massachusetts, August, 1998.

[16] Rivest R. L., Shamir A., and Adleman L. M. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.

[17] Rivest R. T. The MD5 Message Digest Algorithm. *Internet RFC 1321*, April 1992.

[18] C. P. Schnorr. Efficient signature generation by smart cards. *Journal of cryptology*, 4(3):161–174, 1991.

[19] Wang H., Zhang Y. A Protocol for Untraceable Electronic Cash. In Hongjun Lu and Aoying Zhou, editor, *Proceedings of the First International Conference on Web-Age Information Management*, volume 1846 of *Lectures Notes in Computer Science*, pages 189–197, Shanghai, China, 2000. Springer-Verlag.

[20] Wang H., Zhang Y. Untraceable Off-line Electronic Cash Flow in E-Commerce. In *Proceedings of the 24th Australian computer science conference ACSC2001*, pages 191–198, GoldCoast, Australia, 2001. IEEE computer society.

[21] Yacobi Y. Efficient electronic money. In *Advances in cryptology–Asiacrypt'94*, volume 917 of *Lectures Notes in Computer Science*, pages 153–163. Springer-Verlag, 1995.

[22] Yiannis T. Fair Off-Line Cash made easy. In *Advances in cryptology–Asiacrypt'98*, volume 1346 of *Lectures Notes in Computer Science*, pages 240–252. Springer-Verlag, 1998.

[23] Yiannis T., Yung M. On the security of ElGamal-based encryption. In *International Workshop on Practice and Theory in Public Key Cryptography (PKC '98)*, volume 1346 of *Lectures Notes in Computer Science*, Yokohama, Japan, 1998. Springer-Verlag.